PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-222399

(43)Date of publication of application : 21.08.1998

(51)Int.Cl. G06F 11/28
G06F 11/28

(21)Application number : 09-026846

(22)Date of filing : 10.02.1997

(71)Applicant : NEC CORP

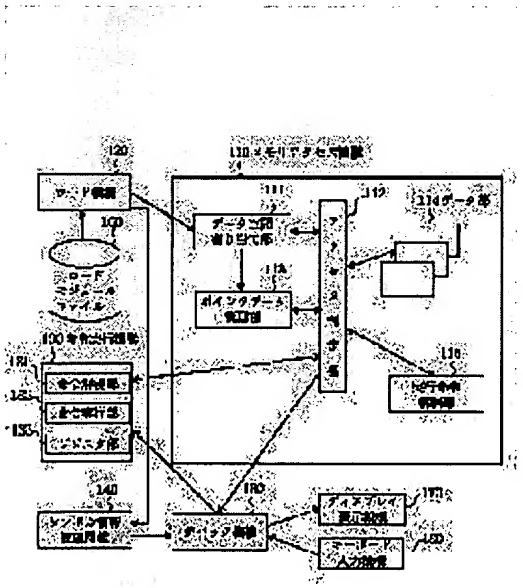
(72)Inventor : SATO TAMOTSU

(54) SIMULATION SYSTEM FOR SIMULATION DEBUGGER

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a simulation debugging system capable of detecting program error by preventing erroneous execution caused by unauthorized address access by monitoring access for every data.

SOLUTION: Allocation information of respective pieces of data are stored in a data space allocation part 111 and the address of pointer data and a data name to be accessed are stored in a pointer data management part 113. When access to the memory is executed at the time of executing the program by an instruction execution mechanism 130, an access judgment part 112 judges whether a register in a same value relation with pointer data is similar to a register executing access to the memory. Then, data accessed by the data name registered in the pointer data management part 12 is taken out from the data space allocation part 111. The range of the memory occupied by data is compared with the address which is access-requested and access violation is detected.



LEGAL STATUS

[Date of request for examination] 10.02.1997

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]	
[Date of final disposal for application]	
[Patent number]	3106989
[Date of registration]	08.09.2000
[Number of appeal against examiner's decision of rejection]	
[Date of requesting appeal against examiner's decision of rejection]	
[Date of extinction of right]	

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]
[Claim 1] The case where the program of a switching system is debugged using a simulation debugger, It is the simulation method of the simulation debugger for detecting access to mistaken memory data, and unjust access out of room. Said simulation debugger is equipped with the data space quota section, the pointer data Management Department, and the access judging section. To the definition data unit inputted into said simulation debugger, to the false space of said simulation debugger Assign room according to an individual through said data space quota section, store a data stereo, and when said data stereo is pointer data Record the initial value of the address of said pointer data, and the address of the accessible data of said pointer data on said pointer data Management Department, and activation of simulation is faced. It judges whether it is pointer data by whether the specified address for access and the address of said access judging section of the pointer data registered correspond. Said access judging section the range of the address corresponding to the specified candidate for access, and the address Call from said pointer data Management Department and said data space quota section, respectively, and when said candidate for access is pointer data The right or wrong of access are judged by whether said pointer data, the register in equivalence relation, and the register that is performing access to memory are the same. The simulation method of the simulation debugger characterized by what the right or wrong of access are judged for by whether furthermore said candidate for access is in the range of said address.
[Claim 2] It is the simulation method of the simulation debugger according to claim 1 which clears registration of the register number of the pointer data which has the same register number when the contents of said pointer data and the register number in the register for simulation activation are registered at said pointer data Management Department in activation of simulation when said candidate for access is pointer data, and the same register number as this register number is already registered into other pointer data, and separates equivalence relation.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]
[0001]
[Field of the Invention] This invention relates to the method which detects access to especially mistaken memory data, and unjust access out of room about the memory access method of the simulation debugger used in case the program of a switching system is debugged.
[0002]
[Description of the Prior Art] In case the program of a switching system is debugged, before carrying a program in a switching system, it is common to use a simulator, in order to perform false transit on a host computer and to perform a check of operation beforehand.
[0003] By the conventional simulator, the false room which the program for actuation uses is stored in the false room which divided into a part for each data division which was assigned by the linkage program, and which prepared by size and was combined by the linkage program, and the run command part, and was assigned.
[0004] Next, a sequential instruction is taken out and performed from a run command part with activation directions. Furthermore, when the memory access for data division breaks out on the occasion of an instruction execution, access to the false room assigned to a part for data division is performed. The performed result is stored in false room and a CPU internal register.
[0005] The contents of false room and the contents of the CPU internal register are displayed on a display by display directions.
[0006] The gestalt of operation of a Prior art is explained to a detail with reference to a drawing. Drawing 7 is the block block diagram of the simulation debugging method of the gestalt of a Prior art, and is equipped with the keyboard entry device 380 in which directions are inputted into the memory access device 310, the load device 320, the instruction-execution device 330, the symbol information management device 340, the load module file 350 that stores a load module, the debugging device 360, the display display device 370 which displays an activation result, and a simulator.
[0007] The memory access device 310 has data division 314 and the run command storing section 315, the load device 320 has the low data input device 321 in which data division and the run command section are inputted, and the symbol information input device 322, and the instruction-execution device 330 has the instruction-decode section 331, the instruction-execution section 332, and the register section 333.

[0008] With the load directions inputted from the keyboard entry device 380, a part for the low data division loaded to a simulator and a symbol information part are read from the program stored in the load module file 350, a part for low data division is passed to the memory access device 310, and the load device 320 passes a symbol information part to the symbol **** control mechanism 340, respectively.
[0009] A part for low data division is divided into a run command part and a data definition by the load device 320, and is divided and stored in the run command storing section 315 and data division 314 of the memory access device 310.
[0010] With the activation directions inputted from the keyboard entry device 380, the instruction-execution device 330 is moved from the debugging device 360, and the memory address and CPU internal register information which decode an instruction by the read-out instruction-decode section 331 serially, and are needed for activation in the run command storing section 315 of the memory access device 310 to an instruction are extracted. When the instruction which needs memory access is decoded, the memory access device 310 is operated and it accesses to the data used as the purpose in false room.

[0011] The specified address value confirms whether to be within the limits of the room assigned to the data division 314 in false room, and if the memory access device 310 is within the limits, it will perform access to the target data. Under the present circumstances, the contents of the range check of access are judged by whether it is within the limits of the quota starting address of data division 314, and an ending address.
[0012]
[Problem(s) to be Solved by the Invention] If it is within the limits, even if it will access data other than the target data accidentally, it is that the mistake is undetectable, in order to perform only the check of whether the address of the data is within the limits of the quota starting address of data division 314, and an ending address, in case the first trouble of the gestalt of the above-mentioned conventional operation performs access to data.
[0013] Although the reason gathers the data of a program in the data division 314 which the linkage program assigned, it is because the false room for size of data division was collectively assigned by the conventional simulator and data have only been arranged in it.

[0014] Since the address in memory was specified accidentally, as a result of changing the contents of data other than the modification purpose, even if the second trouble performs activation which the program mistook, if false room is not crossed, it is that an error is undetectable.
[0015] The reason is because it is not confirming whether be the pointer data which can be accessed for every data according to individual by the conventional simulator.
[0016] The purpose of this invention prevents conventionally activation [made / in according to illegal address access generated since detection was difficult / the mistake] by supervising access for every data, and is to offer the simulation debugging method which can detect the error of a program.
[0017]
[Means for Solving the Problem] The simulation method of the simulation debugger of this invention The case where the program of a switching system is debugged using a simulation debugger, It is the simulation method of the simulation debugger for detecting access to mistaken memory data, and unjust access out of room. A simulation debugger is

equipped with the data space quota section, the pointer data Management Department, and the access judging section. To the definition data unit inputted into a simulation debugger, to the false space of a simulation debugger Assign room according to an individual through the data space quota section, store a data stereo, and when a data stereo is pointer data Record the initial value of the address of pointer data, and the address of the accessible data of pointer data on the pointer data Management Department, and activation of simulation is faced. It judges whether it is pointer data by whether the specified address for access and the address of the access judging section of the pointer data registered correspond. The access judging section the range of the address corresponding to the specified candidate for access, and the address Call from the pointer data Management Department and the data space quota section, respectively, and when the candidate for access is pointer data The right or wrong of access are judged by whether pointer data, the register in equivalence relation, and the register that is performing access to memory are the same, and the right or wrong of access are judged by whether the candidate for access is in the range of the address further.

[0018] In activation of simulation, at the pointer data Management Department, when the contents of the pointer data and the register number in the register for simulation activation are registered when the candidate for access is pointer data, and the same register number as this register number is already registered into other pointer data, it is desirable to clear registration of the register number of pointer data with the same register number, and to separate equivalence relation.

[0019] By the simulator, without adding a hand to a source file, the load module file created by the linkage program can be used, and unlawful access of the data division within a program can be detected.

[0020] The detection at the time of the pointer data in a program accessing data other than the purpose and access of those other than room are detectable.

[0021] Thereby, since access to the inaccurate data division for which detection was difficult is detectable by the conventional simulator, the quality of the program before carrying in the system can be improved.

[0022]

[Embodiment of the Invention] Next, the gestalt of operation of this invention is explained with reference to a drawing.

[1] Explain the configuration of the gestalt of operation of explanation this invention of a configuration to a detail with reference to a drawing. Drawing 1 is the block block diagram of the simulation debugging method of the gestalt of operation of this invention. A memory access device and 111 the sign 110 in drawing The data space quota section, 112 the pointer data Management Department and 114 for the access judging section and 113 Data division, 115 a load device and 130 for the run command storing section and 120 An instruction-execution device, 131 -- the instruction-decode section and 132 -- for a symbol information management device and 150, as for a debugging device and 170, a load module file and 160 are [the instruction-execution section and 133 / the register section and 140 / a display display device and 180] keyboard entry devices.

[0023] Reference of drawing 1 connects the memory access device 110 in which the main role of this invention is played to the load device 120 in which low data are read from a load module file 150, the instruction-execution device 130 in which an instruction is executed, the symbol information management device 140, and the debugging device 160.

[0024] The data for low data division from a load module file 150 are passed to the

memory access device 110 via the load device 120 by the load directions inputted from the keyboard entry device 180. From the load device 120, the data of a symbol information part are also passed at coincidence to the symbol information management device 140. Furthermore, the access address of data is passed to the memory access device 110 from the instruction-execution device 130.
[0025] The instruction-execution device 130 decodes an instruction by the instruction-decode section 131 by reading an instruction from the run command storing section 115 of the memory access device 110 serially with the activation directions inputted from the keyboard entry device 180, the memory address and CPU internal register information which are needed for activation are extracted, and CPU internal register information is memorized at the register section 133. When the instruction which needs memory access for activation is decoded, the access address of data is passed to the memory access device 110 from the instruction-execution device 130.
[0026] The memory access device 110 and the symbol information management device 140 are used for the debugging device 160, and it manages an interface with a user using display display device 170 grade.
[0027] In addition to the data division 114 which were also in the conventional example, and the run command storing section 115, the memory access device 110 has the data space quota section 111, the access judging section 112, and the pointer data Management Department 113.
[0028] The data space quota section 111 assigns data division 114 as false room. The pointer data Management Department 113 manages information, when the data to assign are pointer data. The access judging section 112 judges violation of the memory access from the instruction-execution device 130. Data division 114 are parts which assign the stereo of data. The run command storing section 115 is a part which stores the stereo of a program.
[0029] The data from the load device 120 are passed to the data space quota section 111, and when data are pointer data, they are notified to pointer data ****** 113. The data quota section 111 assigns the address of the data division 114 which became independent one [at a time] to each data containing the passed pointer data.
[0030] The access request from the instruction-execution device 130 is sent to the data space quota section 111 and the access judging section 112 via the access judging section 112, and judges violation of access based on the contents registered into pointer data ****** 113 in the access judging section 112.
[2] Explain explanation of operation, next actuation of the simulation debugging method of the gestalt of operation of this invention to a detail with reference to a drawing. Drawing 2 is the mimetic diagram showing the data flow at the time of reading during the block of the simulation debugging method of the gestalt of operation of this invention, as for a data stereo and D12, data and D16 are program instruction and a data attribute and D13 give [the sign D11 in drawing / the address and D14 / the initial value of data, and D15] the same sign to the same block as drawing 1 . Drawing 3 is a flow chart at the time of reading of the simulation debugging method of the gestalt of operation of this invention, and the signs S101-S112 in drawing are each step.
[0031] The actuation at the time of storing a program in the false room within the memory access device 110 first is explained to a detail. If drawing 1 , drawing 2 , and drawing 3 are referred to, storing will be started by load directions (S101). Reading of low data is performed from a load module file 150 by the load device 120 (S102). (No) and storing will

be ended, if the data stereo D11 and data attribute D12 of the low data read when the existence of low data was judged and there were low (S103) data (Yes) are transmitted to the data space quota section 111 (S104) and do not have low data (S112).
[0032] In the data space quota section 111, a part for program execution instruction part or a data definition is judged (S105), when the read data stereo D11 is a program execution instruction, it stores program instruction D16 in (Yes) and the run command storing section 115 (S110), and it returns to reading (S102) of the following low data. When the read data stereo D11 is a data definition, (S105-No), Allotment according to individual for storing the data stereo D11 in the data division 114 which are false room is performed (S106). Moreover, based on a data attribute D12, it judges whether a data definition is pointer data (S107). When it is pointer data (Yes), the initial value D14 of the address D13 assigned to the pointer data, and the accessible data address of pointer data to the pointer data Management Department 113 Delivery (S108), In pointer data ****** 113, the address D14 and initial value D15 of the sent pointer data are stored as a pointer data management table (S108). Moreover, via the access judging section 112, the data space quota section 111 stores data D15 in data division 114 with data other than pointer data (S107-No) (S111), and returns to reading (S102) of the following load module file.
[0033] Next, the actuation at the time of detecting violation of memory access is explained to a detail. Drawing 4 is the mimetic diagram showing the data flow at the time of violation detection of address access of the simulation debugging method of the gestalt of operation of this invention, as for a register number and D22, access and D26 are error notifications and access directions and D23 give [the sign D21 in drawing / the data-division address and D24 / the quota address and D25] the same sign to the same block as drawing 1 . Drawing 5 is a flow chart at the time of violation detection of the simulation debugging method of the gestalt of operation of this invention, and the signs S201-S212 in drawing are each step. Drawing 6 is a format of the memory quota table of the gestalt of operation of this invention, and a pointer data management table, (a) is a memory quota table and (b) is a pointer data management table.
[0034] If drawing 1 , drawing 4 , drawing 5 , and drawing 6 are referred to, a judgment will be started corresponding to activation directions (S201), and the memory access directions D22 specified by the CPU internal register number D21 of the register section 133 from the instruction-execution device 130 will be sent to the access judging section 112 (S202). The access judging section 112 judges whether it is access with pointer data by whether it is in agreement with the address D23 of the data division applicable to the register number accessed from the pointer data Management Department 113, and the address of the pointer data which takes out the range D24 of data division (quota address) accessed from the data space quota section 111 (S203), and is registered into the pointer data Management Department 113 (S204). When it is access with pointer data (Yes), it judges whether the pointer data registered into the pointer data Management Department 113, the register in equivalence relation, and the register which is performing access to memory are the same, and is accessible data division (S205). A register is not the same, and if it is not accessible data division (S205-No), it will shift to S211 as violation of access. It judges (S206), when it is registered whether the register number same [a register is the same, and] at S205 if judged with their being accessible data division (Yes) as the register number of the register specified as the instruction is already registered into other pointer data, it clears the register of pointer data with (S206-Yes) and the same register number, separates equivalence relation (S207), and it shifts to S208. When not

registered, it shifts to (S206-No) and S208 directly. In S208, the register number of a register and the contents of PONTA data which were specified as the instruction are registered to the pointer data Management Department 113, and it shifts to S209. When it is not pointer data access, it shifts to S209 as (S204-No) and a general data access.
[0035] In S209, it judges [by which it was taken out from the data space quota section 111] whether it assigned and the range of the address of the data division 114 for access is crossed from the address D24 (S209). The access D25 to data division 114 is allowed noting that there are not (Yes) and violation of access, when it is not over the address range (S210), and a judgment is ended (S212). When it is over the address range, it is detected as (S209-No) and violation of access, and shifts to S211. In S211 detected as violation of access, an error notification D26 is sent to the debugging device section 160.
[0036]
[Example]
[1] Explain the configuration of one example of the gestalt of operation of explanation this invention of the configuration of an example to a detail. The load device 120 is connected to the memory access device 110, and the data d1 containing pointer data p1 and program instruction i1 are contained in the low data inputted into the memory access device 110 from the load module file 150. The memory access device 110 is elsewhere connected to the instruction-execution device 130, the symbol information management device 140, and the debugging device 160.
Actuation of one example of the gestalt of operation of explanation this invention of actuation of [2" example is explained to a detail. The actuation at the time of storing a program in the false room within the memory access device 110 first is explained to a detail.
[0037] Reference of drawing 1 , drawing 2 , and drawing 3 transmits each D11 and data "stereo data 1" data attribute "char array" D12 which reading of the data d1 which contain pointer data p1 from a load module file 150, and program instruction i1 was performed (S102), and were read by the load device 120 to the data space quota section 111 (S104).
[0038] In the data space quota section 111, it is judged by data attribute "char array" D12 whether it is program instruction (S105). When it is program instruction i1, program instruction i1 (D16) is stored in (Yes) and the run command storing section 115 (S111). Allotment for storing the data stereo "data1" D 11 in (No), and the false room 0x30000 - 0x3ffff, when it is the data division of a program is performed. Data d1 are assigned to the address 0x30000 to 0x30004 of data in this case (S106).
[0039] Next, based on data attribute "char array" D12, it judges whether the data stereo "data1" D 11 is pointer data (S107).
[0040] When it is pointer data p1, the address 0x30100 (D13) and the accessible data-address initial value 0x30000 (D14) of pointer data are passed to pointer data ****** 112 (S108). In pointer data ****** 112, it carries out a stack, using accessible data-address initial value 0x30000 (D14) of the address 0x30100 (D13) of the sent pointer data, and pointer data as a pointer data management table (S109).
[0041] Moreover, the data space quota section 111 stores the stereo "data1" (D15) of data in the predetermined address 0x30000 to 0x30004 of the false space where data division 114 were assigned in the pointer data with data other than pointer data (S107-No), for example, the address, via the access judging section 112 (S111), and returns to reading (S102) of the following load module file.
[0042] Next, the actuation at the time of detecting violation of memory access is explained

to a detail. If drawing 1 , drawing 4 , drawing 5 , and drawing 6 are referred to, a judgment will be started corresponding to activation directions (S201), and the memory access directions D22 specified with the value 0x30100 (D21) of the CPU internal register r1 of the register section 133 will be sent to the access judging section 112 from the instruction-execution device 130 (S202). The access judging section 112 takes out the range 0x30000 to 0x30004 (D24) of the address 0x3000 of data division to access as indicated to be the address initial value 0x3000 (D23) of the data division applicable to the register number r1 (0x30100) to access as shown in drawing 6 (b) from the pointer data Management Department 113 to drawing 6 (a) from the room quota table of the data space quota section 111 (S203).

[0043] Next, it judges whether it is access with pointer data by whether it is in agreement with the address 0x30100 of the pointer data p1 registered into the pointer data Management Department 113 (S204). When it is access with pointer data (Yes), it judges using the value of the register r1 in which the address 0x30000 which is actually going to perform memory access for whether they are the accessible data division registered into the point data control table of the pointer data Management Department 113 is shown, and the accessible address 0x30000 of the pointer data management table which is in agreement with the register name r1 (S205). A register is not the same, and if it is not accessible data division (S205-No), it will shift to S211 as violation of access. It judges (S206), when it is registered whether the register number same [a register is the same, and] at S205 if judged with their being accessible data division (Yes) as the register number r1 of the register specified as the instruction is already registered into other pointer data, it clears the register of pointer data with (S206-Yes) and the same register number, separates equivalence relation (S207), and it shifts to S208. When not registered, it shifts to (S206-No) and S208 directly. In S208, it registers to the pointer data Management Department 113 by making into equivalence relation the register number r1 of a register and the address 0x30100 of the PONTA data p1 which were specified as the instruction, and shifts to ** (S208) S209.

[0044] When it is not pointer data access, it shifts to S209 as (S204-No) and a general data access.

[0045] In S209, it judges [by which it was taken out from the data space quota section 111] whether it assigned and the range 0x30000 to 0x30004 of the address of the data division 114 for access is crossed from the address D24 (S209). The access D25 to data division 114 is allowed noting that there are not (Yes) and violation of access, when it is not over the address range (S210), and a judgment is ended (S212).

[0046] When it is over the address range, it is detected as (S209-No) and violation of access, and shifts to S211. In S211 detected as violation of access, an error notification D26 is sent to the debugging device section 160.

[0047]

[Effect of the Invention] Since the 1st effectiveness of this invention can discover the part which has caused the violation of a data access in a program, it is that the trouble of a program is solvable at an early stage. That reason is the point which violation of a data access cannot discover most easily among the troubles of a program, and is because it can discover certainly at the time of simulation assigning every one point of this to data division for every data, and by managing pointer data.

[0048] The 2nd effectiveness is being able to shorten the time amount which uses the exchange. It is because the trial which the **** can detect violation of a data access

beforehand, without using the exchange, and uses the exchange can be performed smoothly.

[Translation done.]